IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicants: Jonathan S. Wolf, Arthur B. Mellor, Wayne F. Tackabury, Christopher B. Anderson, Robin M. Whitworth, Michael D. Haag and Brian A. Del Vecchio

Attorney Docket No.: 3186.1000-001

Title: NETWORK CONFIGURATION MANAGER

Date: 6-8-01

EXPRESS MAIL LABEL NO. EL552576037US

PRELIMINARY AMENDMENT

Assistant Commissioner for Patents
Washington, D.C. 20231

Sir:

This Preliminary Amendment is being filed concurrently with the Application. No new matter has been introduced.

Please amend the application as follows:

In the Specification

Please replace the paragraph at page 8, lines 8 through 9 with the following paragraph:

Fig. 6 is a flowchart illustrating the three main processes of the policy-based configuration management of an embodiment of the present invention.

Please replace the paragraph at page 27, lines 15 through 22 with the following paragraph:

The "Customer" subtarget level 610 is broken down further into two subtarget levels. The first of these subtarget levels is a "Customer Routing" subtarget level 612 for configuring a customer's routing information. This could include, for example, setting up static routes, and deciding what routing information is to be advertised to the customer. The second of these subtarget levels is a "Customer Connectivity" subtarget level 614 for which a customer's connectivity information is configured. This can include, for example, setting up permanent virtual circuits (PVCs) to the customer.

Please replace the Abstract at page 40 with the following paragraph:

A policy engine generates configlets that are vendor-neutral, vendor-specific or both, based on a selected target level and a selected device/device group. A translator translates and combines the configlets to form vendor-dependent configuration files. The policy engine generates the configlets using policies associated with the selected target level and its sub-target levels, as defined by a target level hierarchy. A policy includes at least a condition, and an action which the policy engine performs if the condition is true. In performing the action, the policy engine typically writes to at least a partial configlet. A policy may further include a verification clause, which is used to verify a running configuration. Policy dependencies may also be defined such that where a second policy is dependent on a first policy, the second policy must be evaluated after the first policy. This is necessary, where, for example, the first policy generates and stores a value to be used by the second policy. Policies are small programs. A configlet hierarchy is defined, such that a child configlet inherits properties which it does not define from its parent. A mapping function maps infrastructure data in a first format to a second format, so that the second format is recognizable by the policy engine. A loader batches, schedules and loads a configuration file to its intended device. Upon replacing a first device using a first configuration format with a second device using a second configuration format, the first device's

configuration is read in or uploaded and reverse-translated into configlets. The configlets are then translated into a configuration formatted for the second device. The system retains device logins and passwords in encrypted format. A user desiring to connect to a device must log in to the system instead. The system in turn logs in or connects to the device and passes information back and forth between the user and the device, as if the user were logged directly into the device.

Amendments to the specification are indicated in the attached "Marked Up Version of Amendments" (pages i - ii).

Respectfully submitted,

HAMILTON, BROOK, SMITH & REYNOLDS, P.C.

By _____

Gerald M. Bluhm
Registration No. 44,035
Telephone (781) 861-6240
Facsimile (781) 861-9540

Lexington, Massachusetts 02421-4799
Dated: 6/8/01

MARKED UP VERSION OF AMENDMENTS

Specification Amendments Under 37 C.F.R. § 1.121(b)(1)(iii)

Replace the paragraph at page 8, lines 8 through 9 with the below paragraph marked up by way of bracketing and underlining to show the changes relative to the previous version of the paragraph.

Fig. 6 is a flowchart illustrating the ~~two~~ three main processes of the policy-based configuration management of an embodiment of the present invention.

Replace the paragraph at page 27, lines 15 through 22 with the below paragraph marked up by way of bracketing and underlining to show the changes relative to the previous version of the paragraph.

The "Customer" subtarget level 610 is ~~further~~ broken down further into two subtarget levels. The first of these subtarget levels is a "Customer Routing" subtarget level 612 for configuring a customer's routing information. This could include, for example, setting up static routes, and deciding what routing information is to be advertised to the customer. The second of these subtarget levels is a "Customer Connectivity" subtarget level 614 for which a customer's connectivity information is configured. This can include, for example, setting up permanent virtual circuits (PVCs) to the customer.

Replace the Abstract at page 40 with the below paragraph marked up by way of bracketing and underlining to show the changes relative to the previous version of the paragraph.

A policy engine generates configlets that are vendor-neutral, vendor-specific or both, based on a selected target level and a selected device/device group. A translator translates and combines the configlets to form vendor-dependent configuration files. The policy engine generates the configlets using policies associated with the selected target level and its sub-target levels, as defined

by a target level hierarchy. A policy includes at least a condition, and an action which the policy engine performs if the condition is true. In performing the action, the policy engine typically writes to at least a partial configlet. A policy may further include a verification clause, which is used to verify a running configuration. Policy dependencies may also be defined such that where a second policy is dependent on a first policy, the second policy must be evaluated after the first policy. This is necessary, where, for example, the first policy generates and stores a value to be used by the second policy. Policies are small programs ~~written as small programs~~. A configlet hierarchy is defined, such that a child configlet inherits properties which it does not define from its parent. A mapping function maps infrastructure data in a first format to a second format, so that the second format is recognizable by the policy engine. A loader batches, schedules and loads a configuration file to its intended device. Upon replacing a first device using a first configuration format with a second device using a second configuration format, the first device's configuration is read in or uploaded and reverse-translated into configlets. The configlets are then translated into a configuration formatted for the second device. The system retains device logins and passwords in encrypted format. A user desiring to connect to a device must log in to the system instead. The system in turn logs in or connects to the device and passes information back and forth between the user and the device, as if the user were logged directly into the device.